Diagno

Medical Diagnostic Bot

A Python-based solution for preliminary medical analysis

Project by :

AIT2409115 Calvin Christofan Ng AIT2409124 Darrell Benedict Setiawan AIT2409105 Fernando Hartono

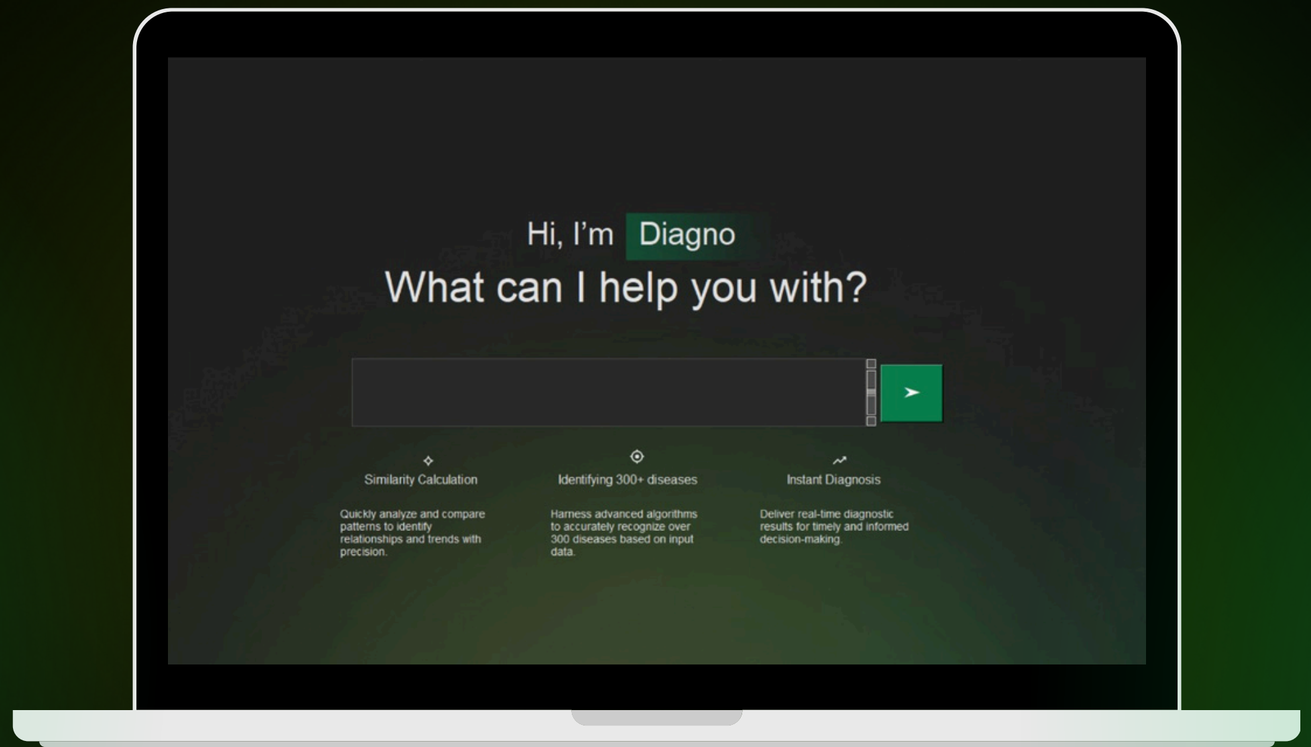
AIT2409116 Filbert Ng AIT2409103 Fidelius William Gandisaputra AIT2409098 Galen Maximilian Boediharto

Aim of the Project

- Identify possible illnesses based on symptoms.
- Provide quick preliminary insights.
- Encourage seeking professional medical advice.

Inner Drive

- Limited access to medical professionals in some areas.
- Increasing need for technology-assisted healthcare solutions.

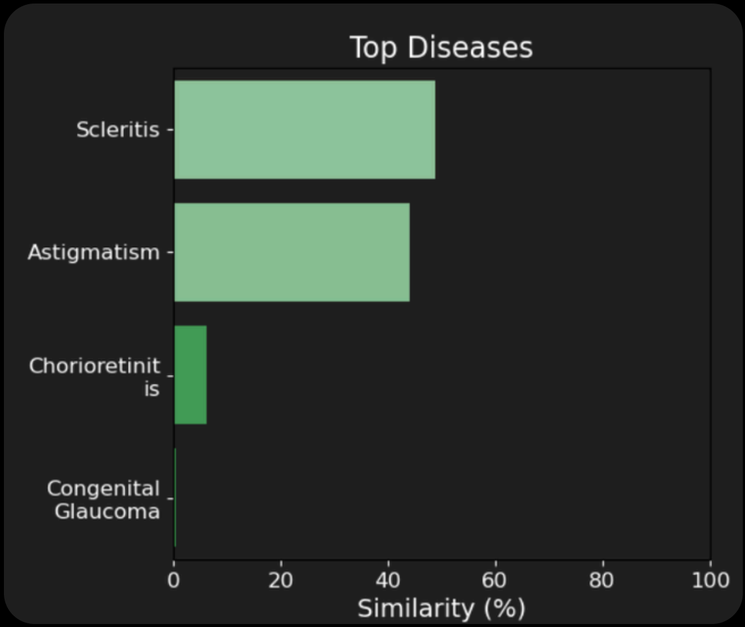


AIT2409105

User enters their issues regarding their illness that contain a list of symptoms through a user-friendly interface.

my eyes are irritated and my vision is blurred.

Displays the top 4 possible diseases



A brief description for the disease

Inflammation of the choroid and retina, often due to infections like toxoplasmosis or autoimmune conditions, causing vision disturbances.

List of symptoms

- Eye pain
- blurred vision
- sensitivity to light

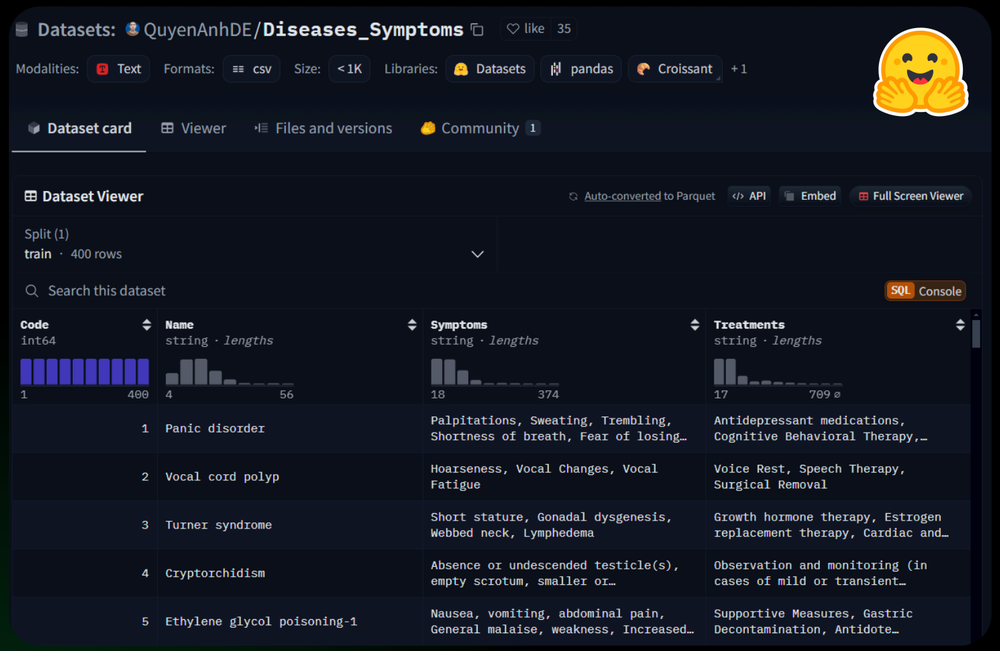
Suggested treatment

- Antibiotics or antivirals for infectious causes
- Corticosteroids to reduce inflammation
- Immunosuppressive therapy for autoimmune-related cases.

AIT2409115

Dataset

Augmenting data & scraping



https://huggingface.co/datasets/QuyenAnhDE/Diseases_Symptoms

	name	Symptoms	General Description	Treatment
0	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	A type of anxiety disorder characterized by su...	Cognitive-behavioral therapy (CBT). Medication...
1	Vocal cord polyp	Hoarseness, Vocal Changes, Vocal Fatigue	Noncancerous growths on the vocal cords caused...	Voice therapy to reduce strain. Surgical remov...
2	Turner syndrome	Short stature, Gonadal dysgenesis, Webbed neck...	A chromosomal disorder in females causing shor...	Growth hormone therapy to improve height. Estr...
3	Cryptorchidism	Absence or undescended testicle(s), empty scro...	A condition in which one or both testicles fai...	Orchiopexy surgery to correct the condition. H...
4	Ethylene glycol poisoning-1	Nausea, vomiting, abdominal pain, General mala...	Toxicity caused by ingesting antifreeze, leadi...	Administration of fomepizole or ethanol as ant...
...	...	...	...	...
395	Urinary Stones (Kidney Stones)	Severe abdominal or back pain, blood in urine,...	Hard mineral deposits that form in the kidneys...	Drink plenty of fluids to help pass the stone....
396	Osteoporosis	Fragile bones, loss of height over time, back ...	A condition that weakens bones, making them fr...	Take calcium and vitamin D supplements. Use me...
397	Rheumatoid Arthritis	Joint pain, stiffness, swelling, fatigue, loss...	An autoimmune disease that causes chronic infl...	Use disease-modifying antirheumatic drugs (DMA...
398	Type 1 Diabetes	Frequent urination, Increased thirst, Weight loss	An autoimmune condition where the body attacks...	Administer insulin therapy. Monitor blood suga...
399	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	A chronic condition where the body becomes res...	Follow a healthy diet and exercise regimen. Ta...

400

400.000

	Code	Name	Symptoms	Symptom Sentences
0	1	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	I experience Fear of losing control, Trembling...
1	1	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	I experience Palpitations, Trembling, Shortnes...
2	1	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	I experience Shortness of breath, Sweating, an...
3	1	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	I experience Sweating, Dizziness, Fear of losi...
4	1	Panic disorder	Palpitations, Sweating, Trembling, Shortness o...	I experience Dizziness, and sometimes Palpitat...
...	...	...	...	...
399995	400	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	I experience Slow healing of wounds, and somet...
399996	400	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	I experience Fatigue, Increased hunger, and so...
399997	400	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	I experience , and sometimes Slow healing of w...
399998	400	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	I experience Increased hunger, Fatigue, and so...
399999	400	Type 2 Diabetes	Fatigue, Increased hunger, Slow healing of wounds	I experience Slow healing of wounds, and somet...

400000 rows × 4 columns

Preprocess Word

Input Example

I experience Fear of losing control, Trembling,

Shortness of breath, Dizziness, and sometimes Sweating.

Cleaning Text

I experience Fear of losing control Trembling Shortness

of breath Dizziness and sometimes Sweating

Casefolding Text

i experience fear of losing control trembling shortness

of breath dizziness and sometimes sweating

Tokenizing Text

['i', 'experience', 'fear', 'of', 'losing', 'control', 'trembling',

'shortness', 'of', 'breath', 'dizziness', 'and', 'sometimes', 'sweating']

Delete Frequent Word

['i', 'fear', 'of', 'losing', 'control', 'trembling', 'shortness', 'of',

'breath', 'dizziness', 'and', 'sweating']

Filtering Text

fear losing control trembling shortness breath dizziness sweating

Lemmatizing Text

fear losing control trembling shortness breath dizziness sweating

AIT2409124

Splitting

Split the dataset into 80% for training and 20% for testing

```
x_train = train['text_lemmatizing']
y_train = train['Code']
x_test = test['text_lemmatizing']
y_test = test['Code']
```

Tokenizer

```
tokenizer = tf.keras.preprocessing.text.Tokenizer()
```

```
tokenizer.fit_on_texts(x_train)
```

```
x_train = x_train.astype(str).tolist()
x_test = x_test.astype(str).tolist()
```


```
early\\\": 463, \\\"satiety\\\": 463, \\\"vomiting\\\": 13816, \\\"bloating\\\": 3771, \
"pain\\\": 93130, \\\"difficulty\\\": 27004, \\\"gripping\\\": 441, \\\"using\\\": 949, \
"bruising\\\": 7757, \\\"deformity\\\": 2816, \\\"small\\\": 3158, \\\"painful\\\": 6647,
```

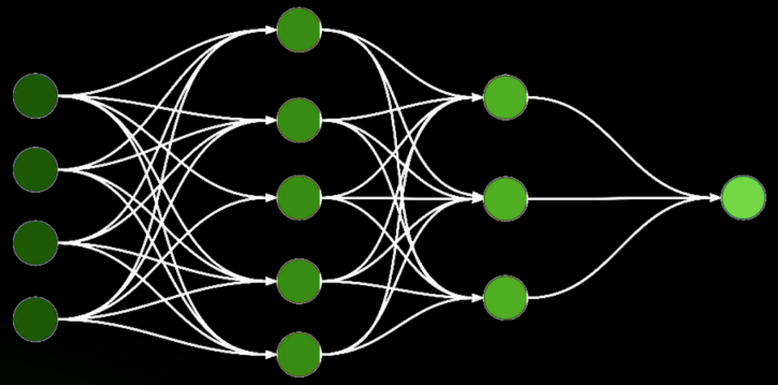

Padding

```
# Convert x_train and x_test into sequences
x_train_seq = tokenizer.texts_to_sequences(x_train)
x_test_seq = tokenizer.texts_to_sequences(x_test)

# Pad the sequences to ensure they have same length
max_len = max(len(seq) for seq in x_train_seq)
x_train_seqs_padded = pad_sequences(x_train_seq, maxlen=max_len, padding='post')
x_test_seqs_padded = pad_sequences(x_test_seq, maxlen=max_len, padding='post')
```

```
x_train shape: (320000, 35)
y_train shape: (320000,)
x_val shape: (80000, 35)
y_val shape: (80000,)
```

AIT2409116

Model



```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 35)	0	-
embedding (Embedding)	(None, 35, 300)	271,800	input_layer[0][0]
bidirectional (Bidirectional)	(None, 35, 512)	1,140,736	embedding[0][0]
attention (Attention)	(None, 35, 512)	0	bidirectional[0]… bidirectional[0]…
bidirectional_1 (Bidirectional)	(None, 256)	656,384	attention[0][0]
batch_normalization (BatchNormalizatio…	(None, 256)	1,024	bidirectional_1[…
dense (Dense)	(None, 512)	131,584	batch_normalizat…
dropout (Dropout)	(None, 512)	0	dense[0][0]
dense_1 (Dense)	(None, 256)	131,328	dropout[0][0]
dropout_1 (Dropout)	(None, 256)	0	dense_1[0][0]
dense_2 (Dense)	(None, 401)	103,057	dropout_1[0][0]

Hyperparameter Tuning

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=1e-6)
model_checkpoint = ModelCheckpoint('trial.keras', save_best_only=True)
```

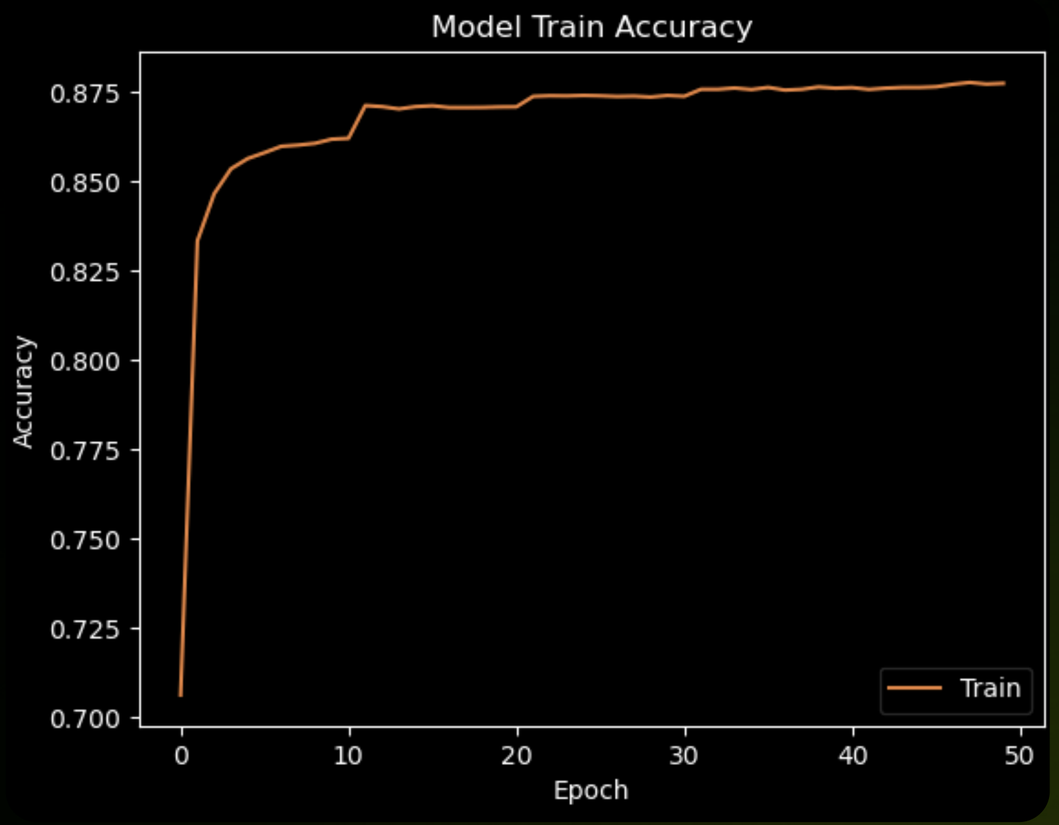
AIT2409105

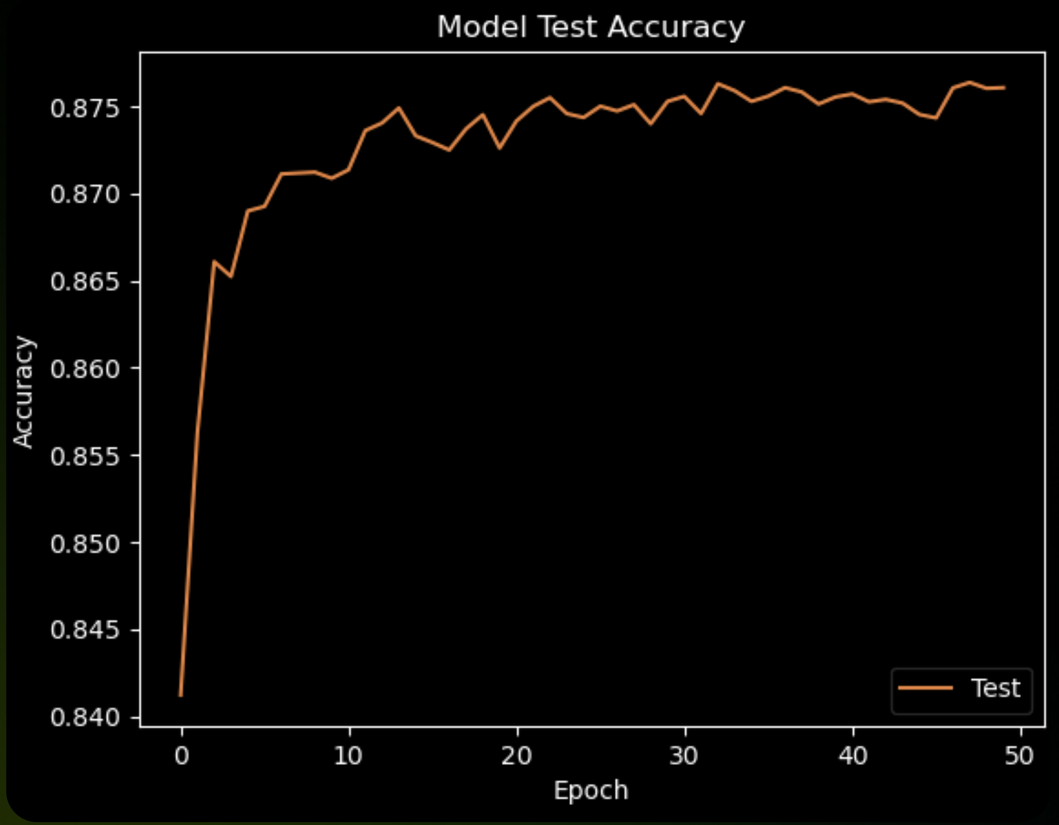
Training

```python
# Using tf.data for efficient handling of large datasets
train_dataset = tf.data.Dataset.from_tensor_slices((x_train_seqs_padded, y_train))
train_dataset = train_dataset.shuffle(buffer_size=10000).batch(128).prefetch(tf.data.AUTOTUNE)

val_dataset = tf.data.Dataset.from_tensor_slices((x_test_seqs_padded, y_test))
val_dataset = val_dataset.batch(128).prefetch(tf.data.AUTOTUNE)

# Model training
history = model.fit(
    train_dataset,
    validation_data=val_dataset,
    epochs=50,
    callbacks=[early_stopping, reduce_lr, model_checkpoint]
)
```

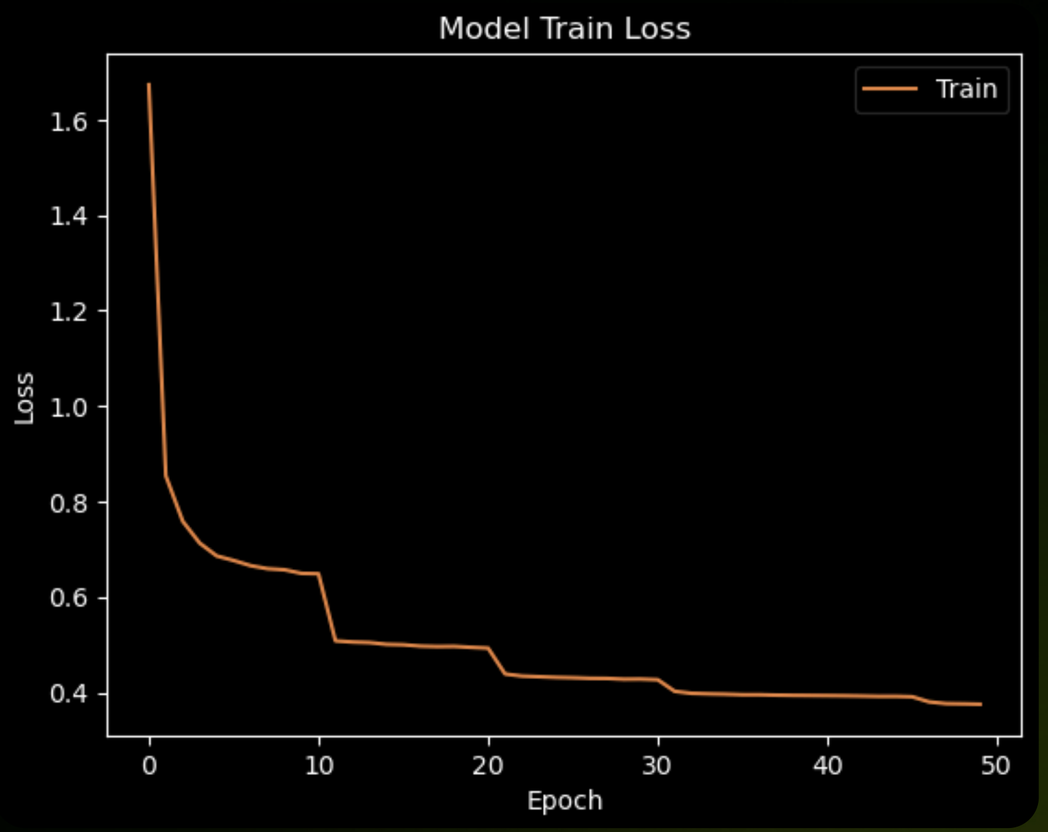
```
Epoch 1/50
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 524s 207ms/step - accuracy: 0.5242 - loss: 2.9978 - val_accuracy: 0.8412 - val_loss: 0.8416 - learning_rate: 0.0010
Epoch 2/50
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 491s 196ms/step - accuracy: 0.8275 - loss: 0.8909 - val_accuracy: 0.8563 - val_loss: 0.7215 - learning_rate: 0.0010
Epoch 3/50
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 492s 197ms/step - accuracy: 0.8447 - loss: 0.7711 - val_accuracy: 0.8661 - val_loss: 0.6715 - learning_rate: 0.0010
Epoch 4/50
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 493s 197ms/step - accuracy: 0.8520 - loss: 0.7193 - val_accuracy: 0.8652 - val_loss: 0.6584 - learning_rate: 0.0010
Epoch 5/50
```

Accuracy Plotting





Loss Plotting



AIT2409103

Mapping

```
{1: 'Panic disorder',                  11: 'Eye alignment disorder',
 2: 'Vocal cord polyp',                12: 'Headache after lumbar puncture',
 3: 'Turner syndrome',                 13: 'Pyloric stenosis',
 4: 'Cryptorchidism',                  14: 'Adenoid cystic carcinoma',
 5: 'Ethylene glycol poisoning-1',     15: 'Pleomorphic adenoma',
 6: 'Ethylene glycol poisoning-2',     16: 'Warthin tumor',
 7: 'Ethylene glycol poisoning-3',     17: 'Mucoepidermoid carcinoma',
 8: 'Atrophic vaginitis',              18: 'Acinic cell carcinoma',
 9: 'Fracture',                        19: 'Mucocele',
10: 'Cellulitis',                      20: 'Osteochondrosis',
```

```python
mapping = {}
for i in range(0, 400000, 1000):
    mapping[preprocessed_df['Code'][i]] = preprocessed_df['Name'][i]
print(mapping)
```

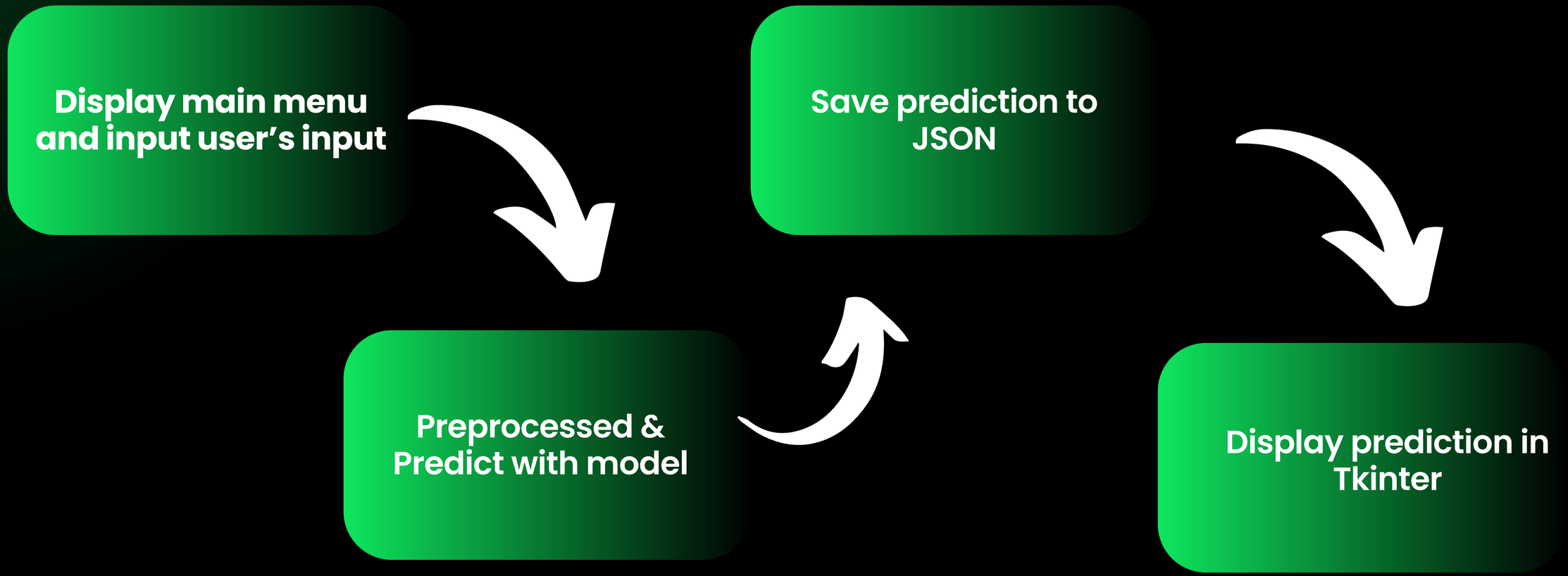
Evaluating

```python
model_output = new_model.predict(x_test_seqs_padded)

# Test the entire validation data (x_test)
for sample_idx, single_output in enumerate(model_output):
    single_output = single_output / np.sum(single_output)

    # Get the top 4 indices and probabilities
    top_4_indices = np.argsort(single_output)[-4:][::-1]
    top_4_probs = single_output[top_4_indices]
    top_4_percentages = top_4_probs * 100

    print(f"Input : {x_test[sample_idx]}")
    for i, (index, prob) in enumerate(zip(top_4_indices, top_4_percentages)):
        class_name = mapping.get(index, f"Class {index}")
        print(f"  Rank {i + 1}: {class_name} - Similarity: {float(prob):.2f}%")
    print()
```

```
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 83s 33ms/step
Input : visual field defect
  Rank 1: Low-Tension Glaucoma - Similarity: 50.24%
  Rank 2: Normal-Tension Glaucoma - Similarity: 49.74%
  Rank 3: Pituitary Adenoma - Similarity: 0.00%
  Rank 4: Labyrinthitis - Similarity: 0.00%

Input : growth conjunctiva white part eye
  Rank 1: Pterygium - Similarity: 100.00%
  Rank 2: Anorexia Nervosa - Similarity: 0.00%
  Rank 3: Abdominal Hernia - Similarity: 0.00%
  Rank 4: Anemia of Chronic Disease - Similarity: 0.00%

Input : fever shortness breath fatigue
  Rank 1: Vasculitis - Similarity: 35.10%
  Rank 2: Abscess of the Lung - Similarity: 33.08%
  Rank 3: Pulmonary Eosinophilia - Similarity: 26.26%
  Rank 4: Pneumonia - Similarity: 1.96%

Input : loss appetite difficulty swallowing coughing wheezing
  Rank 1: Foreign Body in the Gastrointestinal Tract - Similarity: 100.00%
  Rank 2: Cirrhosis - Similarity: 0.00%
  Rank 3: Peritonsillar Abscess - Similarity: 0.00%
  Rank 4: Magnesium Deficiency - Similarity: 0.00%
...
  Rank 2: Carpal Tunnel Syndrome - Similarity: 0.02%
  Rank 3: Adenoid cystic carcinoma - Similarity: 0.00%
  Rank 4: Lung Contusion - Similarity: 0.00%
```

AIT2409103

Program Flow



AIT2409115

Conclusion

Diagno *is an advanced machine learning model designed to analyze user-inputted symptoms in natural language (RNN) and predict the top 4 potential diseases based on the similarity.*

Future Improvements :

- Adding more diseases and symptoms to the database.
- Enhancing the algorithm for better accuracy.
- Implementing multi-language support or voice input.

AIT2409124 AIT2409116